PD-201125

Method and System for Authenticating a User in a Web-based Environment

Technical Field

[0001] The present invention relates generally to a web-based access system, and more particularly, to a system for providing authentication of a user for a web-based system.

Background

[0002] Providing services such as educational services through the Internet or other distributed network system requires potential users of the system to be authenticated by the server that hosts the desired applications. Corporations and other users that have their own web servers may desire services from a central source or server. One way in which a user may use services is to connect directly to the server. However, many corporations have enterprise firewalls that do not allow the potential user to log directly into the central server having the desired application.

[0003] Kerberos is a security protocol typically used for single login to an enterprise system for accessing services from multiple servers in the enterprise. Kerberos is typically used on physically insecure networks and is based on a key distribution model. It allows entities communicating over the networks to prove their identity to each other while preventing eavesdropping or replay attacks. It also provides a

data stream integrity and secrecy using cryptography systems. Kerberos works by providing principals with tickets that they can use to identify themselves to other principals and secret cryptographic keys for secure communication with other principals. A ticket is typically a sequence of a few hundred bytes. These tickets can be imbedded in virtually any other network protocol, thereby allowing the processes implementing that protocol to be sure about the identity of the principals involved. Kerberos is mostly used in application-level protocols such as Telnet or FTP, to provide user to host security.

[0004] It is well known that Kerberos security protocol does not work for Windows 95 and 98 environments because the file system is not secure enough and does not allow multiple users. Also, the Kerberos protocol is not designed for hypertext transfer protocol environments.

[0005] It would therefore be desirable to provide a system that uses some Kerberos-based principals to allow users to communicate through their firewalls to reach remote applications.

Summary of the Invention

[0006] It is therefore one object of the invention to provide an improved method and system for authenticating a user in a web-based environment.

[0007] In one aspect of the invention, system and method for authenticating a client having a privilege server, a head end server, and a web adapter perform the steps of negotiating an authentication scheme between

the server proxy and the privilege server. User information is presented to the web adapter. The user information is provided to the head end server and in turn presents the information to the web adapter. The user is validated in accordance with the authentication scheme. When the user is validated a ticket is generated for the user. The ticket is presented to the client privilege server proxy which decrypts the ticket. A token is formed from the ticket and the client user identification. The token from the client is provided to the privilege server. A packet is formed having a sequence number and session key encrypted with the ticket. The packet is provided to the head end server which in turn authenticates the user. The packet is provided to the client privilege proxy that decrypts the packet and sends the ticket and the sequence number encrypted with the session key to the data server through the web adapter. User is validated at the data server and privileges are granted thereto.

[0008] In a further aspect of the invention, a method for accessing a service comprising:

presenting a ticket and sequence number to a service through the web adapter;

choosing a service in the service server;

sending the session name encrypted with the ticket and user identification to the privilege server and requesting a session key and sequence number;

receiving the session name from the user;

validating the user ticket and privilege;

when the user is validated, issuing the session key and sequence number for the ticket;

encrypting the session key and sequence number with the ticket to from a packet; and

sending the packet and ticket to the service.

[0009] One advantage of the invention is that the system is capable of operating through firewalls and coupling the user to various servers within the firewall. Another advantage of the invention is that the system is capable for use with multiple users.

[0010] Other advantages and features of the present invention will become apparent when viewed in light of the detailed description of the preferred embodiment when taken in conjunction with the attached drawings and appended claims.

Brief Description Of The Drawings

[0011] Figure 1 is a block diagrammatic view of an authentication system according to the present invention.

[0012] Figure 2 is a block diagrammatic view of the privilege server of Figure 1.

[0013] Figure 3 is a flow chart of a method for an initial login to the system.

[0014] Figure 4 is a flow chart of a negotiation for an authentication scheme.

[0015] Figure 5 is a flow chart of accessing a service.

[0016] Figure 6 is a flow chart for registering a user.

[0017] Figure 7 is a flow chart illustrating changing a password.

[0018] Figure 8 is a flow chart for logging a user out of the system.

Description Of The Preferred Embodiment

[0019] In the following figures the same reference numerals will be used to identify the same components in the various views.

[0020] Referring now to Figure 1, an authentication system 10 is illustrated coupled to a user 12 through a user privilege server proxy 14. User privilege server proxy 14 has a dataline 16 that couples the user privilege server proxy 14 to authentication system 10. Dataline 16 may, for example, be an Internet connection through various routers and the like as is known to those skilled in the art. Dataline 16 may be a physical wired connection or wireless connection such as a satellite. Although only one user 12 is illustrated, the present invention is suitable for multiple users.

[0021] Authentication system 10 has a web adapter 18 coupled to a head end server 20. Head end server 20 has a head end server privilege proxy 22. Head end server 20 is coupled to a privilege server 26. Privilege server 26 has a policy engine 28 therein. Authentication system 10 authenticates user 12 so that the user may be coupled to user services 30. As illustrated, user services 30 may include various service servers such as a data server 32 having a data server privilege server proxy 34. User services may also include a server gateway (Serge) 36 and a server

gateway privilege server proxy 38. Both the data server privilege server proxy 34 and the server gateway privilege server proxy 38 are coupled to privilege server 26. Of course, those skilled in the art will recognize various types of services that may be enabled by authentication system 10 beyond those shown in Figure 1.

[0022] Web adapter 18 is implemented partially in hardware and partially in software. In hardware, physical connections to dataline 16 are formed. Web adapter 18 receives the information from the user and forwards the information to head end server 20. Head end server 20 is also implemented partially in hardware and software. Head end server 20 provides an interconnection or functions as an intermediate server between web adapter 18 and privilege server 26. Privilege server 26 will be further described below but is also implemented at least partially in software.

[0023] Referring now to Figure 2, privilege server 26 is illustrated in further detail. Policy engine 28 is the main component within privilege server 28. Policy engine 28 is coupled directly to various functional blocks such as a proxy coordinator 40, a storekeeper 42, an obfuscate/deobfuscate block 44, and key generator 46. Policy engine 28 interacts with proxy coordinator 40 to either publish or subscribe to messages or requests between various proxies in the system. Policy engine 28 interacts with storekeeper to retrieve and update persistent information. Policy engine 28 interacts with obfuscate/deobfuscate block 44 to obfuscate or deobfuscate the data being either sent to the user or

received from the clients. Policy engine 28 interacts with key generator 46 to issue either a ticket or sequence number for a user. Policy engine 28 decides which services of which components should be used in what sequence.

[0024] The policy engine 28 is formed by privilege server 26. Policy engine 28 creates the proxy coordinator 40, storekeeper 42, obfuscate/deobfuscate block 44, and key generator 46 upon startup. As part of the storekeeper function the policy engine also creates and loads user information lists and session information lists. The policy engine also sets the authorization information after the authorization scheme is completed. Policy engine 28 also gets the credentials that are initiated by the client for a particular session. Each of these functions will be further described below.

[0025] Proxy coordinator 40 is illustrated coupled to a data server privilege server proxy block 48, a head end server privilege server proxy block 50, and a SERGE privilege server proxy block 52. These blocks represent information used to communicate to the privilege server proxies rather than the actual privilege server proxies themselves. The proxy coordinator 40 is a component used to communicate with the privilege server proxies. The coordinator 40 helps propagate credentials of the user in sessions to a specific proxy identified by a proxy identification. The proxy coordinator uses a publish/subscribe mechanism for data propagation. The proxy coordinator maintains a list of proxy IDs and their respective interfaces in block 48, 50, and 52.

Any proxies wanting to listen or send instructions within the system must join proxy coordinator 40.

[0026] Storekeeper 42 is coupled to various blocks including a key distribution center block 54, a relational database block 56, a Microsoft active directory block 58, and a lightweight directory access protocol block 60, and another functions block 62. Key distribution center block 54 is a key distribution center of the Kerberos system. The function of this block will be further described below. Database block 56 is a relational database that includes a user information list 64 and session information list 66. User information list 64 is a list of active users in the system and their associated credentials. All information about the user with respect to the privileges, mainly the current ticket, start time and end time of the ticket as well as the active session of the user are in the user information list. Database also includes a session information list 66 that is used for maintaining the session specific credentials in the system. All the session related information with respect to the privilege servers such as session ID, the proxy ID and session key are maintained within a list.

[0027] Obfuscate/deobfuscate block 44 obfuscates or deobfuscates the data with the given key. Obfuscate/deobfuscate block 44 is essentially a scrambler or descrambler of the various information. Obfuscate/deobfuscate block 44 may be coupled to various blocks including a common application programming interface block 68 (CAPI) or message digest 5 (MD5) block 70. Obfuscate/deobfuscate block may also be

coupled to other blocks 72 depending on the specific functions of the specific system. Both the function of CAPI block 68 and message digest 5 block 70 are known to those skilled in the art.

[0028] Key generator 46 is responsible for the generation of tickets and sequence numbers as will be further described below. Also, key generator 46 generates session keys for a given session. Tickets may, for example, include various information such as a lifetime indicator, a start time and an end time. The tickets may also include random numbers as part thereof to prevent duplication. Key generator 46 is coupled to Kerberos block 74, a common application programming interface block 76, a message digest 5 block 78, and another function block 80. Kerberos block 74 as mentioned in the background of the invention is a security mechanism typically used for a single log-on to use resources within a trusted network. Some of the general functions of Kerberos are represented by block 74. Blocks 76 and 78 are similar to that of block 68 and 70 above.

[0029] Referring now to Figures 1, 2 and 3, the initial log in sequence for a client or user to the system is illustrated. In step 90, an authentication scheme is negotiated. The negotiation of the authentication scheme will be further described below in Figure 4. Once a negotiation mechanism or token is accepted by the system, the negotiation mechanism is validated while the client is logged into the system. When the client logs out of the system the negotiation

mechanism is invalidated as will be further described below.

[0030] In step 92 the client privilege server proxy 14 presents a user ID and other required information to web adapter 18. Web adapter 18 represents the user and transfers the user ID and required information to head end server 20. Head end server 20 then provides the user ID and required information to privilege server 26.

[0031] In step 94, privilege server 26 validates the user through policy engine 28. Once a user has been validated by policy engine 28, the key generator 46 generates a ticket for that user. The ticket is encrypted with the user password. The ticket is then coupled to head end server 20 which in turns transfers the ticket to web adapter 18. Web adapter 18 transfers the ticket to user privilege server proxy 14. The privilege server proxy 14 can decrypt the ticket only if it has the appropriate password.

[0032] In step 96, after the ticket is decrypted with the user password in step 94, a default service access request token encrypted with a ticket and user ID is transferred back through web adapter 18 and head end server 20 to privilege server 26. Privilege server 26 validates the user and issues a sequence number and session key encrypted with the ticket to form a packet. The packet is provided to head end server 20 which learns that the user is who he claims to be and access to the system is allowed. The head end server 20 passes the packet to privilege server proxy through web adapter 18.

[0033] In step 98, the client server privilege proxy 14 sees the sequence number and key encrypted with the ticket from privilege the server 26 and decrypts the packet with the user password and provides the ticket and sequence number encrypted with the session key to the service desired such as data server 32. In step 100, the user is validated by receiving the packet from web adapter 18. The packet is decrypted with the session key while privilege server proxy 34 validates the user in coordination with privilege server 26. Once the user is validated the data server 32 allows appropriate roll based privileges to be performed by user 12.

[0034] Referring now to Figure 4, the negotiation authentication scheme described above in step 90 is further described in step 102. The client privilege server proxy proposes one security mechanism or an ordered list of security mechanisms to the privilege server 26. A negotiation token may be used to establish the authentication mechanism context. In step 104, if the privilege server authentication model decoding modules are not present within the client machine, the user 12 may try to download the module from a specific URL. The privilege server accepts or rejects step 104. If the token is not accepted in step 106, step 102 is re-executed. If the token is accepted in step 106, the token is stored. Once a mechanism has been selected, tokens specific to the selected mechanism are carried with a security token. The token may, for example, include various information such as a time stamp in step 110. Time stamping a token may include providing a

start time, a stop time, and a duration length of the valid token.

[0035] Referring now to Figures 1, 2 and 5, once authentication has taken place as illustrated in Figure 3, user access to a particular service 30 is provided. As mentioned in Figure 3, the user is provided a session key and sequence number from the privilege server. User 12 presents the ticket and sequence number encrypted with the session key to a service 30 such as data server 32 through web adapter 18.

[0036] In step 114, the client is validated and given a role-based access to data server 32. Sessions may be chosen from an ordered list in step 114. The user privilege server proxy 14 encrypts the session name with the ticket and passes this information to privilege server 26 through head and server 20 and web adapter 18.

[0037] In step 116, when the session name from user 12 is provided to privilege server 26, the user ticket and privileges for the requested session are validated. If the request is valid, privilege server 26 issues a sequence number and session key against the ticket. The sequence number and key is encrypted with a ticket presented by the user in step 118. In step 120, the client proxy decrypts the packet of the session number and session key against the ticket as described above. The sequence number and ticket encrypted with the session key are provided to the desired service 30 in step 120.

[0038] Referring now to Figures 1, 2, and 6, various administration functions may also be performed by system

10. One such function is registering a user. In step 24, a request for user registration is received by privilege server 26. The user sends its user ID and required information based upon the underlying authentication scheme to web adapter 18. Web adapter 18 provides the information to privilege server through head end server 20.

[0039] In step 124, an asymmetric key is issued by privilege server 26. The asymmetric key is forwarded through head end server 20 to web adapter 18. Web adapter 18 provides the asymmetric key to proxy server 14. In step 126, an open connection is for authentication purposes between user privilege server proxy 14 and web adapter 18.

[0040] In step 128, user privilege server proxy 14 presents the password encrypted with the asymmetric key described above from the open connection. The asymmetric key and user ID is provided to head end server 20 which in turn provides the information to privilege server 26. Privilege server 26, in step 130, registers the user ID and password with the policy engine 28.

[0041] Referring now to Figures 1, 2, and 7, password changes are also provided for in the administrative functions of the system. To change a password, the user privilege server proxy 14 presents the old password, a new password, and user identification to privilege server 26 through head end server 20. Preferably, the password and the old password are encrypted with the ticket in step 132. In step 134, privilege server 26 on receiving the old password, new password and user ID

validates the user and issues a new pair of sequence numbers and a session key. These are provided to privilege server proxy 14 through head end server 20 and web adapter 18.

[0042] Referring now to Figures 1, 2, and 8, the administrative functions may also include logging out of the system. The privilege server proxy 14 requests a sequence number and session key from privilege server 26 through head end server 20 and web adapter 18 by sending the head end server access request token encrypted with a ticket and the user ID.

[0043] In step 138, privilege server proxy issues a sequence number and sequence key encrypted with a ticket after validating the user. The head end server 20 is accessed and forwards the information to web adapter 18. Web adapter 18 forwards the information to privilege server proxy 14.

[0044] In step 140, the privilege server proxy receives the encrypted sequence number and presents the ticket, the token and sequence number encrypted with the sequence key to web adapter 18. Web adapter 18 requests a log out from the head end server 20. Head end and server 20 logs the user out in step 142.

[0045] While particular embodiments of the invention have been shown and described, numerous variations and alternate embodiments will occur to those skilled in the art. Accordingly, it is intended that the invention be limited only in terms of the appended claims.